

Nom/prénom :

no camipro :

Solutions - Programmation 2014

Test intermédiaire – série 1a

Durée 45min.

Mettre votre nom/prénom/no camipro sur toutes les pages.

Aucun document, ni appareil électronique (calculatrice, natel, etc.) n'est autorisé.

Pas de feuille volante reprise, uniquement les feuilles distribuées seront reprises.

Si vous devez faire des hypothèses, notez-les.

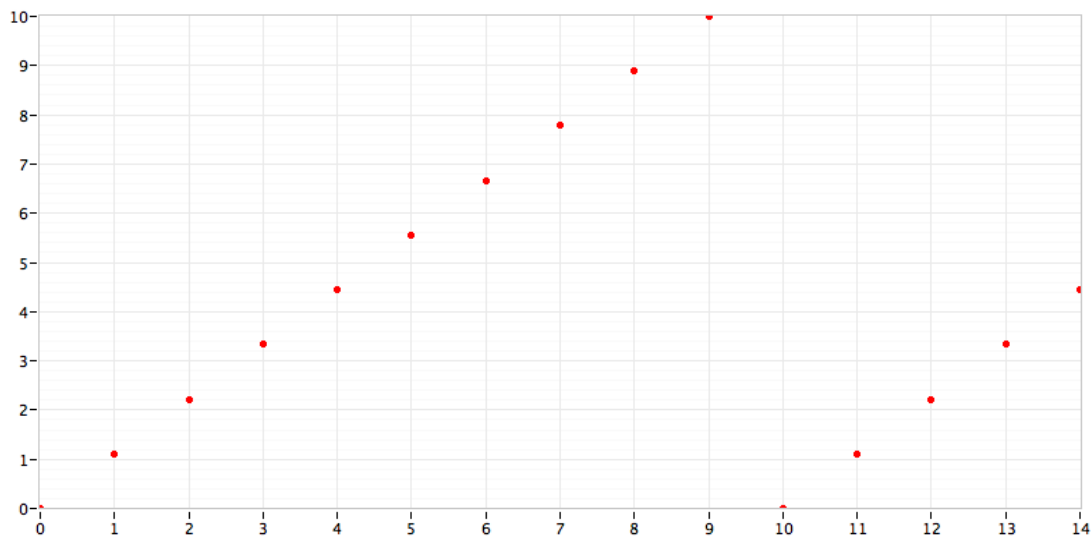
Les autres séries sont des variations de cette série

Nom/prénom :

no camipro :

1. On désire générer un signal périodique triangulaire dont on spécifie le nombre de points **total** n , l'amplitude a et l'offset o . Le nombre de points pour **une** période est fixé à **10**. Les données calculées seront retournées dans un tableau de *double*. La fonction retourne 0 si tout s'est bien passé, ou un code d'erreur dans les autres cas (codes définis ci-après).

Dans l'**EXEMPLE** ci-dessous $n : 15$, $a : 10$ et $o : 0$



Ecrire la fonction **Triangle()** suivante:

```
int Triangle(double points[], int n, double amplitude, double offset)
```

avec n entre [8 et 32]

retourne 0 si pas d'erreur
-1 si n est invalide (pas dans les bornes spécifiées)

Note: vous n'avez pas à vous soucier de la taille de `points[]`.

Vous devez :

- écrire la fonction **Triangle(...)** en c/c++
- gérer les erreurs (n invalide)
- ajouter quelques commentaires pour décrire ce que vous faites dans votre fonction

Nom/prénom :

no camipro :

```
int Triangle(double points[], int n, double amplitude, double offset) {
    const int periode(10); // nb de pt par période
    int err(0);           // pas d'erreur par défaut

    if ((n>=8) and (n<=32)) // dans l'intervalle demandé?
        for (int i=0; i<n; i++) // pour les n pts, modulo periode
            points[i]=(i % periode) * amplitude / (periode-1) + offset;
        else
            err=-1; // interval invalide

    return err;
}
```

Nom/prénom :

no camipro :

2a. Un programmeur débutant écrit le code suivant ?

```
if (a=b)
    cout << "gagné";
else
    cout << "perdu";
```

Si l'entier **a** vaut 5, quelle doit être la valeur de l'entier **b** pour que "perdu" soit affiché ? Justifiez !

L'erreur du programmeur débutant est d'avoir mis '=' à la place de '=='. Il y a donc une assignation au lieu d'une comparaison. Pour afficher "perdu" il faut que l'expression (condition) soit fausse. La convention est: 0-> false, autres valeurs-> vrai
=> b doit être 0

2b. Qu'affiche le code suivant ?

```
int main(){
    int j(2);
    for (int i(j);i<4;i++)
        if (i==3) cout<<"i"<<endl;
        else
            cout << i <<endl;
}
```

Le code ci-dessus peut être réécrit comme :

```
for (int i(2);i<4;i++){ // avec i= 2,3
    if (i==3)
        { cout<< "i" <<endl; }
    else
        { cout << i <<endl; }
}
```

Il affichera:

```
2
i
```

Nom/prénom :

no camipro :

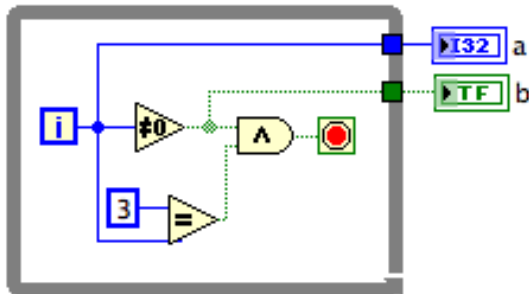
Note: pour les diagrams ci-après, tous les éléments des tableaux sont visibles ; le dernier '0' grisé ne fait pas partie du tableau. **Attention à bien différencier les tableaux des scalaires !** Rappel :

AND 

OR 

Stop if true 

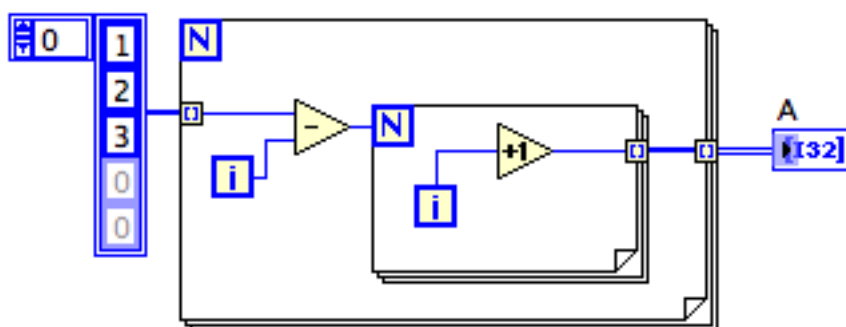
3. Quelles sont les valeurs de a et b après l'exécution du *diagram* ci-dessous?



a : 3
 b : true

Condition d'arrêt avec AND -> 2 fils doivent être vrai.

4. Quelle est la valeur de A après l'exécution du *diagram* ci-dessous?

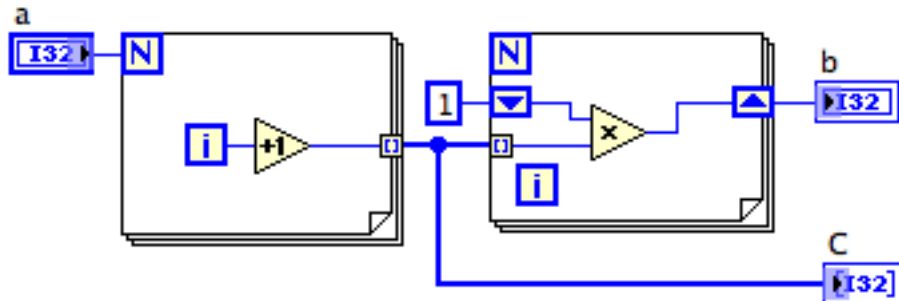


Avec N boucle externe = taille tableau entrée = 3,
Et $N(s)$ boucle interne (1-0), (2-1), (3-2) = 1,1,1
Sortie boucle interne : [1] répété 3 x -> $A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Nom/prénom :

no camipro :

5. Que représente b en fonction de a ? Que vaut C après l'exécution du *diagram* ci-dessous ?



$$C = [1..a]$$

$$b = 1 * C[0] * C[1] * \dots * C[a-1] = C[0] * C[1] * \dots * C[a-1] = 1 * 2 * \dots * a = a!$$